

ID3 Decision Tree

3130160 Nikolaos Papadogoulas

3130198 Archontellis-Rafael Sotirchellis

Το project αυτο υλοποιει τον αλγοριθμο ID3 ο οποίος παραγει ενα δεντρο αποφασης ! Περιλαμβανεται μεσα στο project ενα αρχειο με ονομα train.dat που εχει χρησιμοποιηθει για testing. Μπορειται βεβαια να χρησιμοποιεισετε δικο σας αρχειο (το οποιο βεβαια να ειναι αυτης της μορφης) με οσα attributes θελετε και οσα values το καθε attribute επισης θελετε.

Στον φακελο περιλαμβανεται ξεχωριστα :ολοκληρο το solution , το executable , το source code.

Γραφτηκε με την γλωσσα c# στο VisualStudio με τις βιβλιοθηκες :
using System;
using System.Collections.Generic;

Επειδη η μεθοδος LearnAndCreateTree ειναι αρκετα πολυπλοκη προτεινω για την ορθη κατανοηση της να χρησιμοποιησετε τον debugger του VisualStudio Community 2015 για να εχετε προσβαση σε καθε μεταβλητη και σε καθε Dictionary σε καθε loop της αναδρομης.

Αποτελειται απο 3 κλασεις. Την Node, την Learner και την Data

Η Node ανριπροσοπευει ενα κομβο του δεντρου. Αυτο που εχει σημασια να δουμε σε αυτην την κλαση ειναι οι μεταβλητες της:

- valueOfPrevNode : η μεταβλητη αυτη περιεχει το value του attribute του πατερα κομβου. Χρησιμευει στο να ξερει που να κατευθυνθει ο traverser οταν ερθει η ωρα να διασχισουμε το δεντρο για να εξαγουμε ενα συμπερασμα. Η ριζα επειδη δεν εχει προγονους εχει -1 valueOfPrevNode
- Το attrName κραται το ονομα της ιδιοτητας
- Το result σε ολους τους κομβους ειναι -1 εκτος και αν ειναι φυλλο. Τοτε ειναι 0 η 1 (ναι η οχι)
- Το gain χρησιμευει στην μεθοδο LearnAndCreateTree για να κανει ορθη επιλογη της σειρας καταταξης των κόμβων

Περιλαμβανει επισης μια μεθοδο την ScanAndCalculate η οποια δεχεται δεδομενα και διασχιζει ολο το δεντρο για να εξαγει το συμπερασμα.

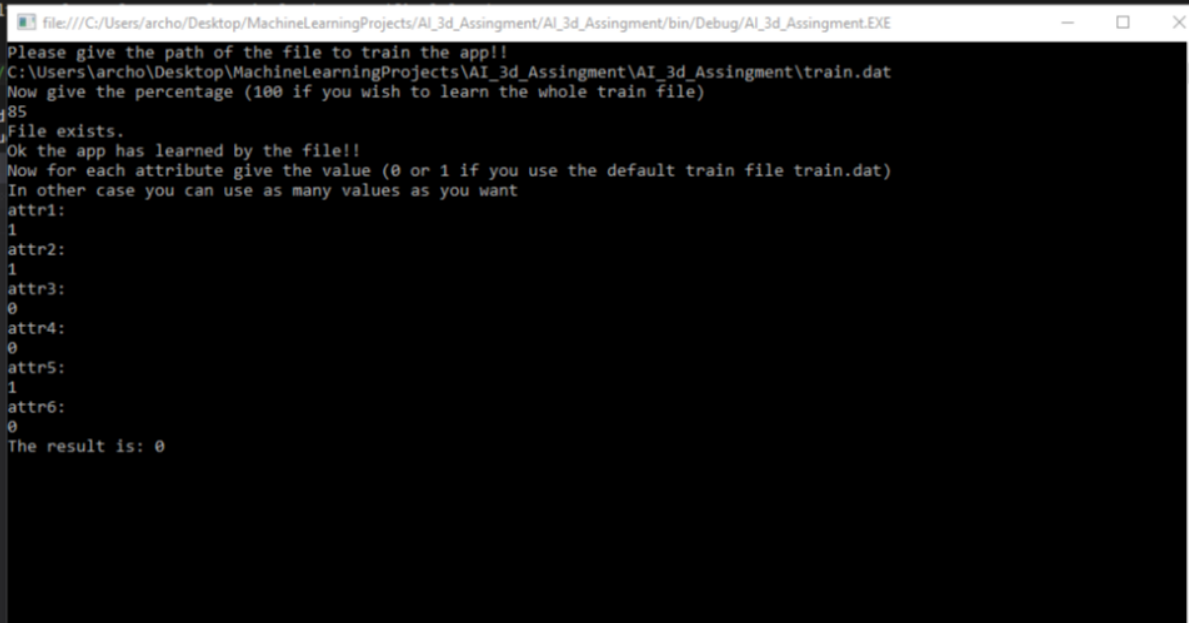
Για την κλάση learner θα περιγράψω μόνο την μεθοδο LearnAndCreateTree καθώς όλες οι υπολοιπες χρησιμοποιούνται απο αυτην.

```
Data.cs Program.cs Node.cs Learner.cs X
ID3 AI_3d_Assingment.Learner LearnAndCreat

1 using System;
2 using System.Collections.Generic;
3
4 /// <summary> 3130160 Nikolaos Papadogoulas 3130198 Archontellis-Rafael Sotirchellis
5
6
7
8
9 namespace AI_3d_Assingment
10 {
11     class Learner
12     {
13         string filename;
14         int percentage;
15
16         public Learner(string filename, int percentage) {...}
17
18         // lets start learning and create the root node
19         public Node Start() {...}
20
21         //this function generates a decision tree recursively
22         public void LearnAndCreateTree(Dictionary<string, int[]> TrainingDictionary, int[] finalClass, Node node, Data data) {...}
23
24         // If all the attributes in final class equals valueToChecked returns True
25         public bool ValidateFinalClass(int[] finalClass, int valueToChecked) {...}
26
27         // This function is used when we cant calculate a pure result but we are in a leaf node ...
28         public int calcResultInCaseOfTerminalAndNotPure(int[] finalClass) {...}
29
30         // Returns the count of positives in final class
31         public int getCountPositives(int[] FinalClass) {...}
32
33         // Calculates the entropy
34         public double getEntropy(int[] array) {...}
35
36         //Log base 2 , need this for entropy
37         public static double log2(double num) {...}
38
39     }
40 }
```

- Αρχικα απο το line 60 μεχρι το 87 ελεγχω αμα ολα τα result ειναι θετικα η αρνητικα .Τοτε πρεπει να σταματησουμε καθως φτασαμε σε leaf node με σιγουρη αποφαση.
- Στο λινε 89 ξεκινω για καθε attribute να βρω το gain του , το οποιο για να βρεθει απαιτει την εντροπια του. Μετραω τα possitives τα neggatives για καθε value του attribute. Τα gain υπολογιζονται στο 136 -162.
- 165-184 παιρνω το attribute με το μεγαλυτρο gain
- 192-255 για καθε μοναδικο value καθε attribute με δοδομενο το attribute με το μεγαλυτερο gain : χρησιμοποιω την μεθοδο split της καλσης data για να απομονωσω τα στοιχεια που εχουν το επιλεγμενο MaxGainAttribute με την συγκεκριμενη τιμη και τα δινω ως δεδομενα στο παιδι που δημιουργειται .Ελεγχω αμεσως μετα αν το παιδι εχει οντως λιγοτερα δεδομενα απο τον γονεα , αν οχι σημαινει οτι δεν μπορουμε να βρουμε 100% ξεκαθαρο 0 η 1 και κανω τον κομβω αυτο leaf δινοντας του ως result το πολυπλυθεστερο αποτελεσμα.
- Τελος καλω αναδρομικα την μεθοδο

Για 800 γραμμές δεδομένων προς training και 6 attributes που περιεχει το αρχείο train.dat ο αλγοριθμος τρεχει σε λιγοτερο απο 1 δευτερολεπτο. Παραθετω και ενα screenshot με τον τροπο χρησης του exe.



```
file:///C:/Users/archo/Desktop/MachineLearningProjects/AI_3d_Assingment/AI_3d_Assingment/bin/Debug/AI_3d_Assingment.EXE
Please give the path of the file to train the app!!
C:\Users\archo\Desktop\MachineLearningProjects\AI_3d_Assingment\AI_3d_Assingment\train.dat
Now give the percentage (100 if you wish to learn the whole train file)
85
File exists.
Ok the app has learned by the file!!
Now for each attribute give the value (0 or 1 if you use the default train file train.dat)
In other case you can use as many values as you want
attr1:
1
attr2:
1
attr3:
0
attr4:
0
attr5:
1
attr6:
0
The result is: 0
```